A Multi-Objective Hierarchical Methodology for Synthesis of Large Scale Electronic Designs

Subtitle: The Top-Down Bottom-Up Methodology (TDBU)

Presented by Trent McConaghy
Analog Design Automation
July 13, 2001
The Third NASA/DoD Workshop on Evolvable Hardware

Presentation Summary

- **Introduction**
- Review of Other Design Methodologies
- Presentation of the TDBU
- Ramifications of the TDBU
- Conclusion

Introduction:
The Problem

- In circuit design, need ways to:
  - Handle massive complexity
  - Minimize design time
  - Minimize number of people needed
  - Maximize design quality – optimal results

Introduction: Methodology Is Needed

- "The benefits of following a disciplined design methodology absolutely outweigh the costs." -Art de Geus, *IEEE Spectrum*, January 2000

Introduction:
Methodology Is Needed

- "A fundamental requirement for success is a clear strategy that coordinates the entire design process" -Henry Chang et al., <u>A Top-Down Constraint-Driven Design Methodology for Analog Integrated Circuits</u>, 1997

Presentation Summary

- Introduction
- **Review of Other Design Methodologies**
- Presentation of the TDBU
- Ramifications of the TDBU
- Conclusion

Review of Other Design Methodologies

- Top-down Constraint-driven Design Methodology

- Bottom-up Design Methodology

- Others: Flat, Concurrent

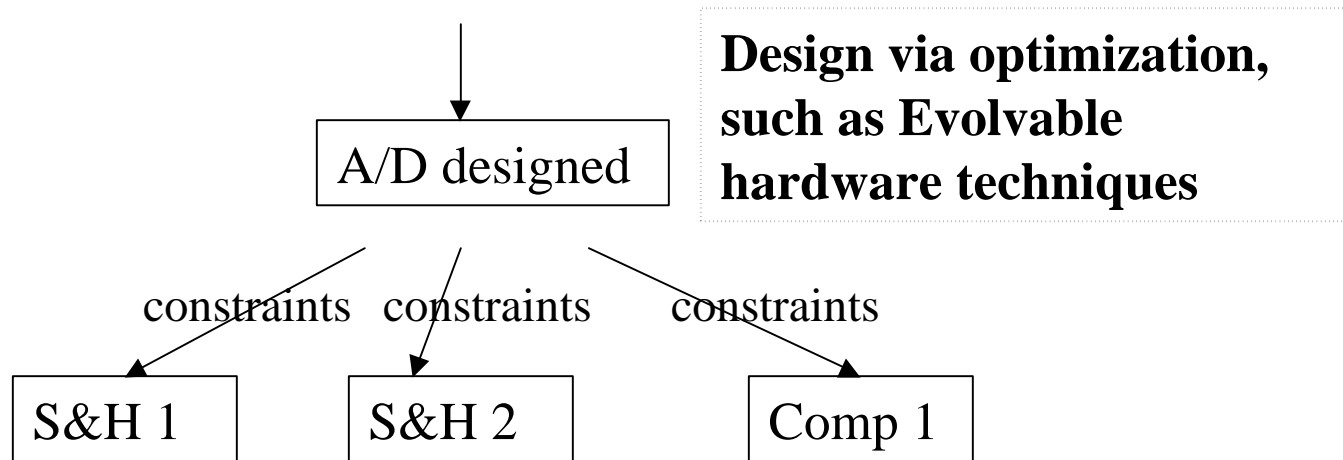- With each methodology, there can be varying levels of automation (eg via EH)

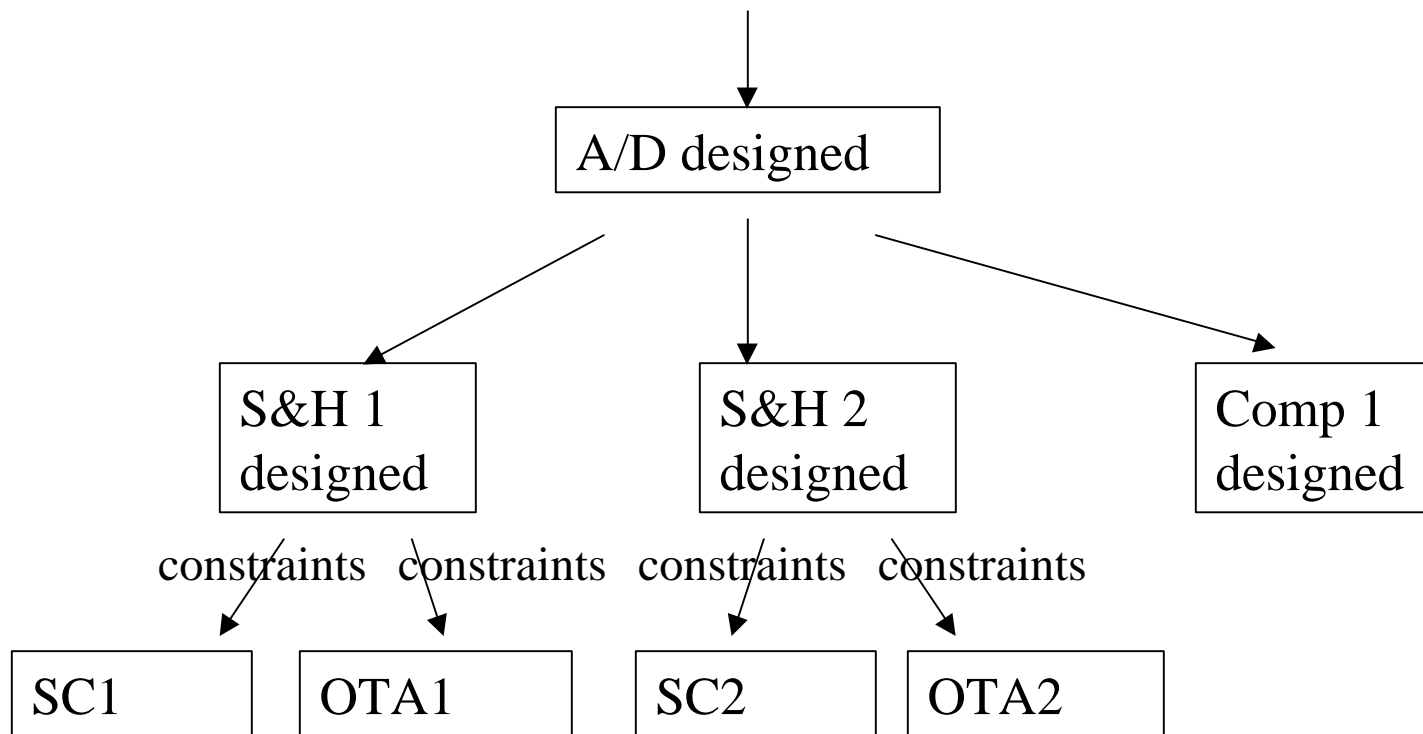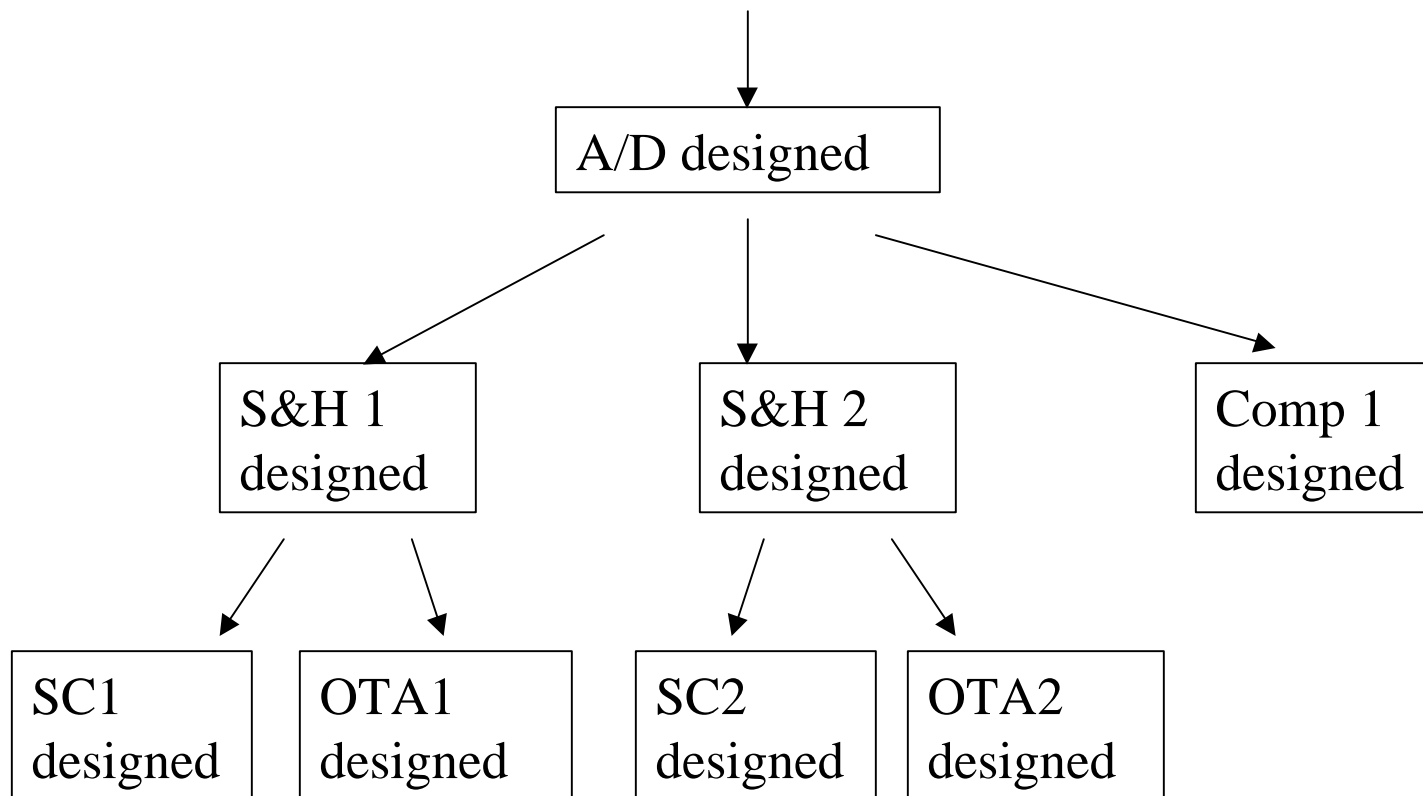# Top-down Constraint-driven Design Methodology

constraints
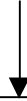
```
      |
      v
+--------------+
| A/D          |
| wanted       |
+--------------+
```

Top-down Constraint-driven Design Methodology

```
                    │
                    ▼
              ┌──────────────┐   ┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐
              │ A/D designed │     Design via optimization,
              └──────────────┘     such as Evolvable
                                   hardware techniques
                                 └ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘

   constraints   constraints        constraints

┌──────────┐   ┌──────────┐       ┌──────────┐
│  S&H 1   │   │  S&H 2   │       │  Comp 1  │
└──────────┘   └──────────┘       └──────────┘
```

Top-down Constraint-driven Design Methodology

```
                    ┌─────────────────┐
                    │  A/D designed   │
                    └─────────────────┘
           ┌──────────────┼──────────────────┐
           ▼              ▼                   ▼
    ┌───────────┐   ┌───────────┐      ┌───────────┐
    │  S&H 1    │   │  S&H 2    │      │  Comp 1   │
    │ designed  │   │ designed  │      │ designed  │
    └───────────┘   └───────────┘      └───────────┘
```

constraints   constraints   constraints   constraints

```
┌─────────┐   ┌──────────┐   ┌─────────┐   ┌──────────┐
│   SC1   │   │   OTA1   │   │   SC2   │   │   OTA2   │
└─────────┘   └──────────┘   └─────────┘   └──────────┘
```

Top-down Constraint-driven Design Methodology

Top-down Constraint-driven Design Methodology
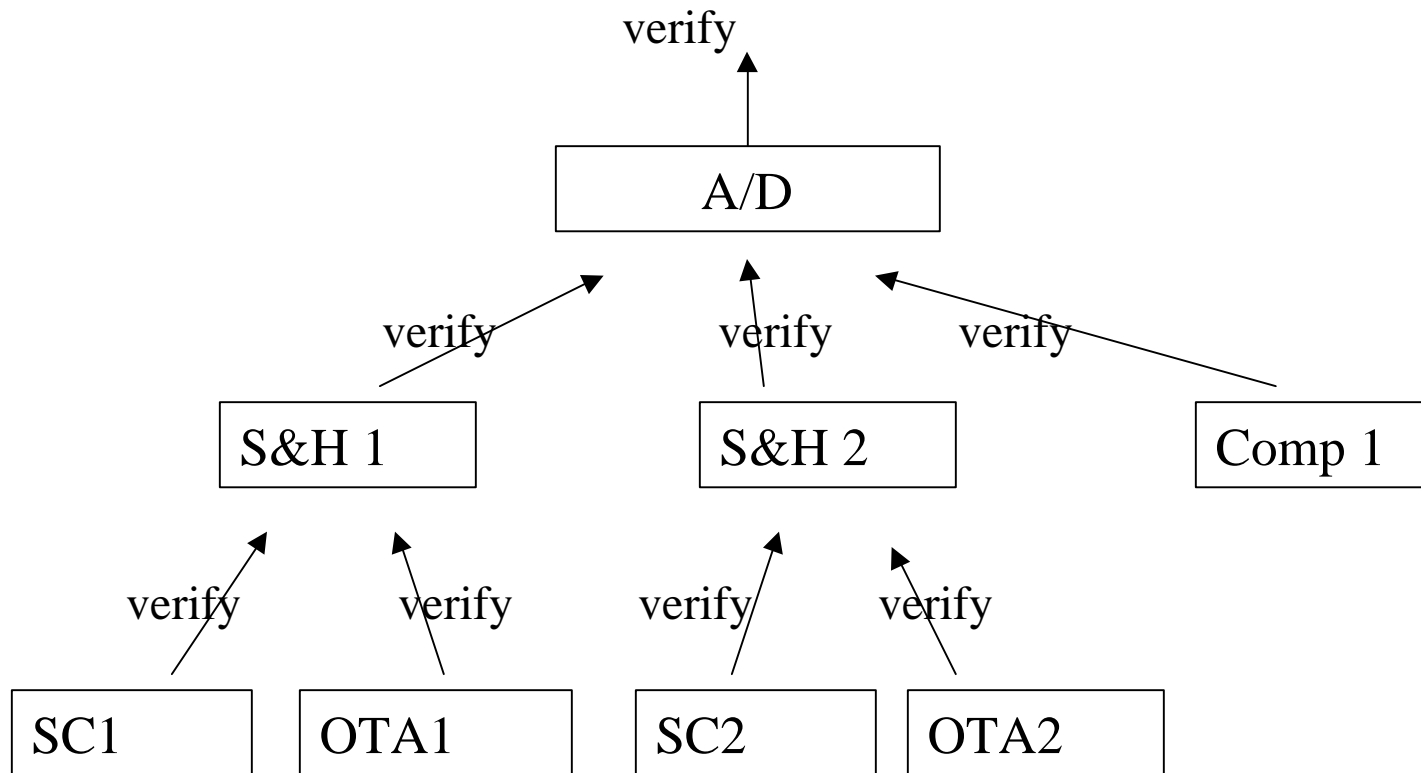
Top-down Constraint-driven Design Methodology
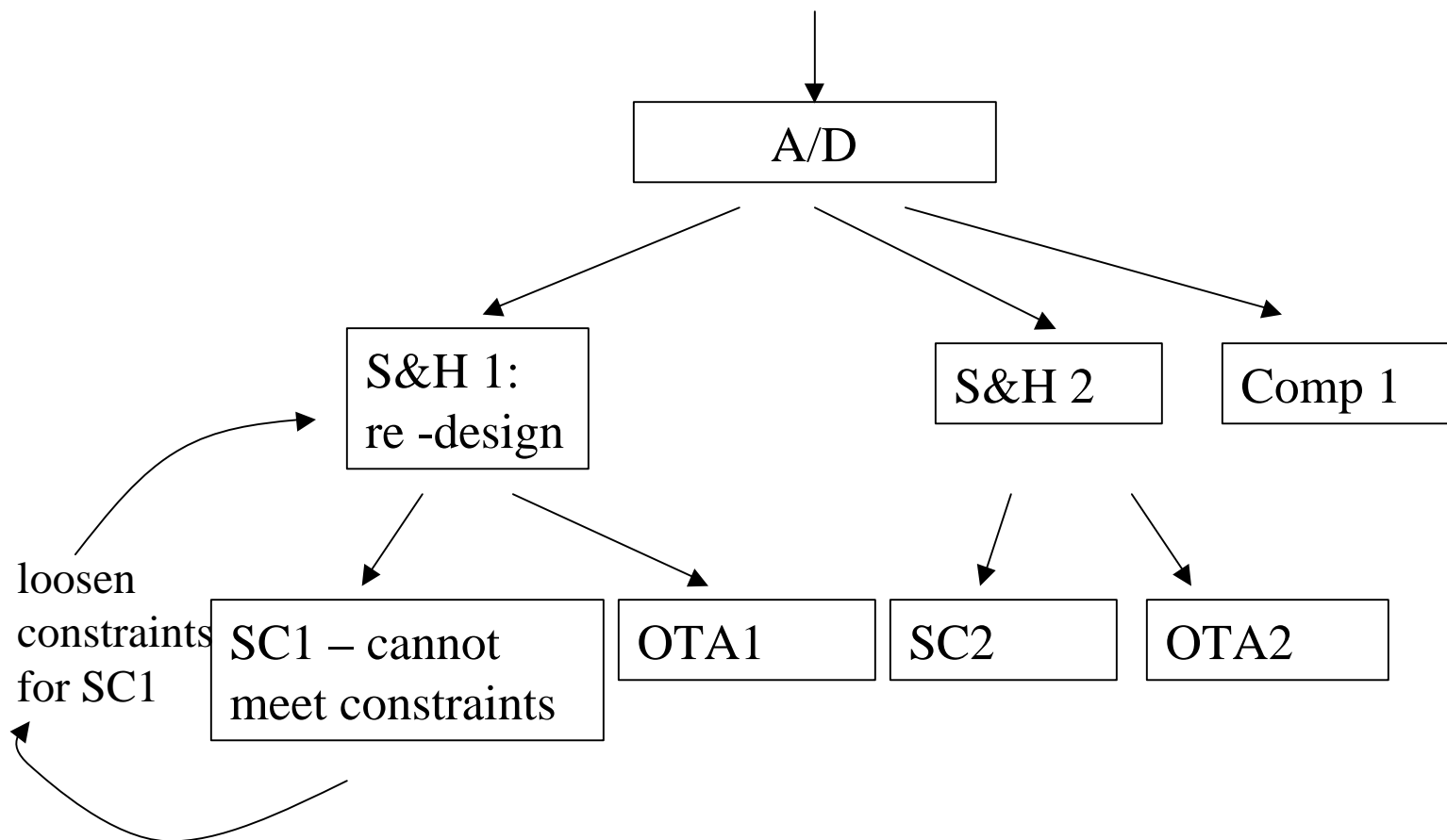
Top-down Constraint-driven Design Methodology

- Advantages:
  - Hierarchical abstraction to manage complexity
  - Can parallelize design efforts
- Disadvantages:
  - Rely on past experience with similar problems to set "reasonable constraints"
  - May have to loosen top-level constraints
  - Iterative up-and-down as constraints get changed
  - Forces architecture selection up-front
  - Designs are not optimal, just feasible

Bottom-up Design Methodology

A/D
wanted

"anticipated
to be
needed"

"anticipated
to be
needed"

"anticipated
to be
needed"

"anticipated
to be
needed"

| SC1 designed & verified | OTA1 designed & verified | SC2 designed & verified | OTA2 designed & verified |

Bottom-up Design Methodology
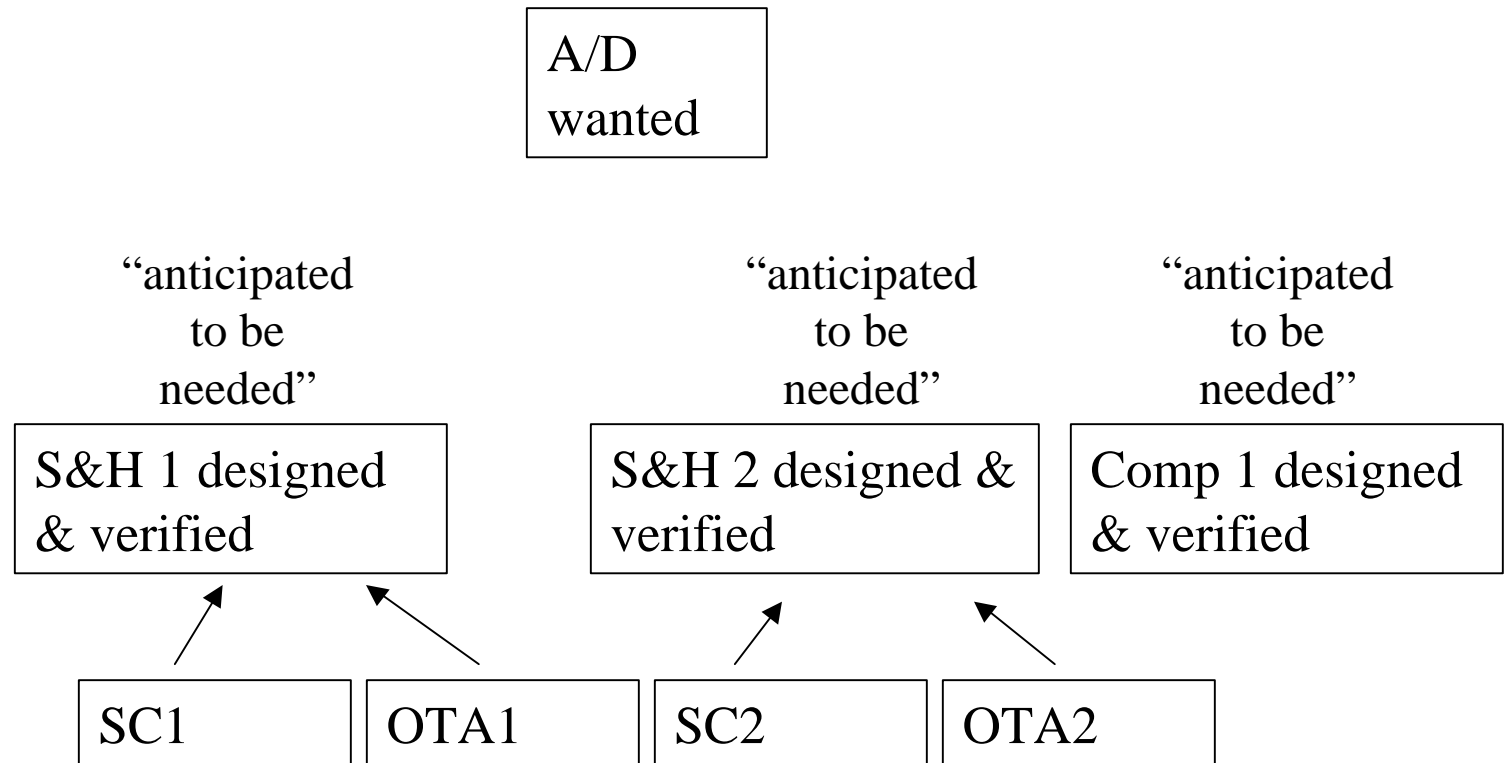
Bottom-up Design Methodology
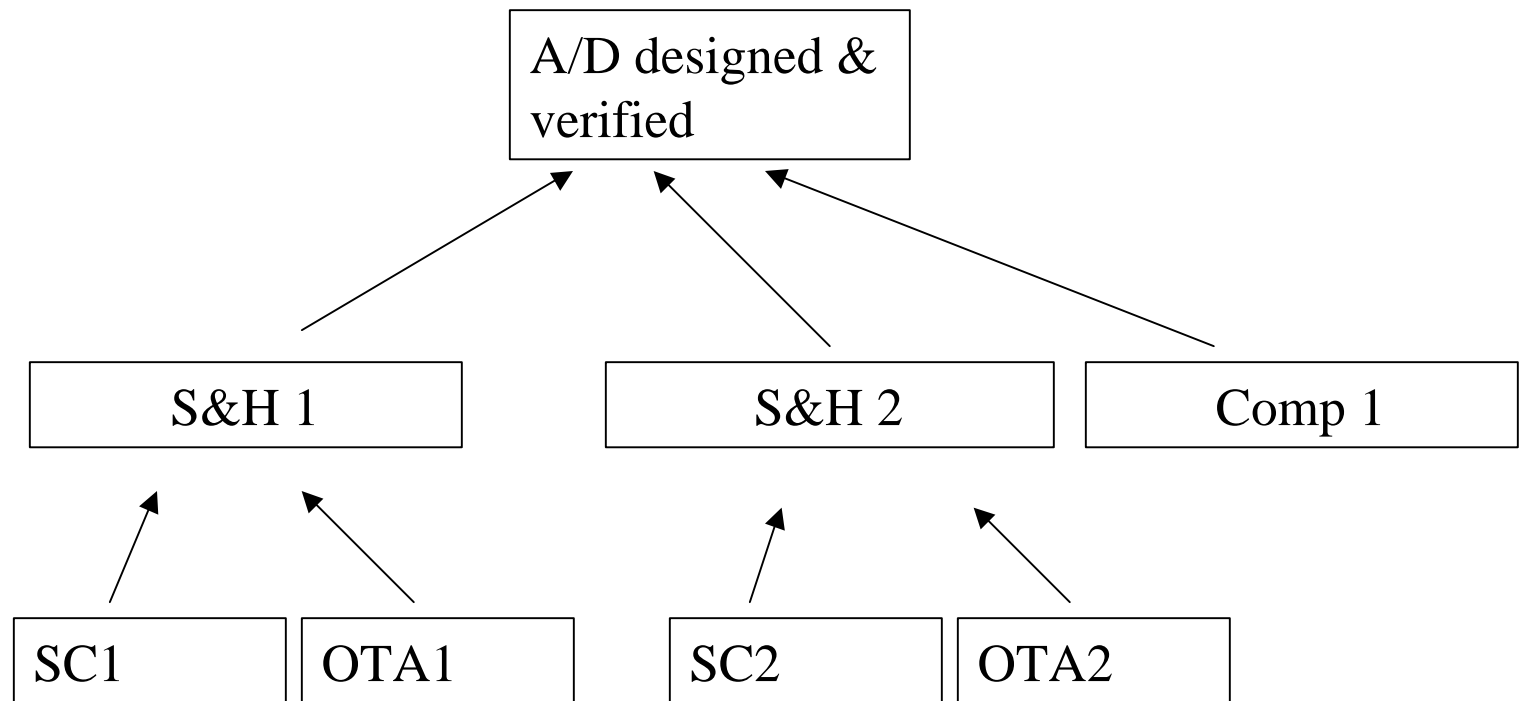
Bottom-up Design Methodology
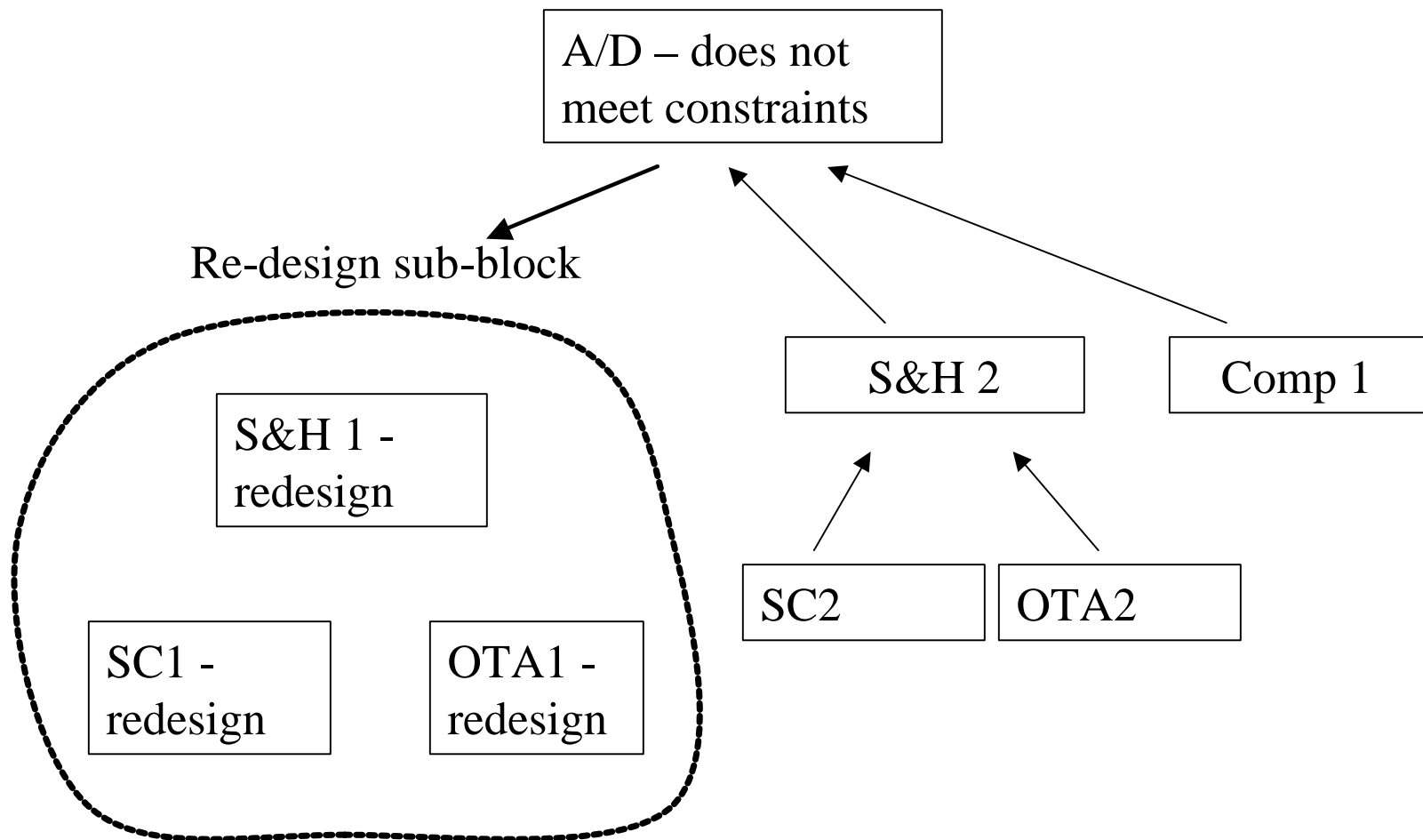
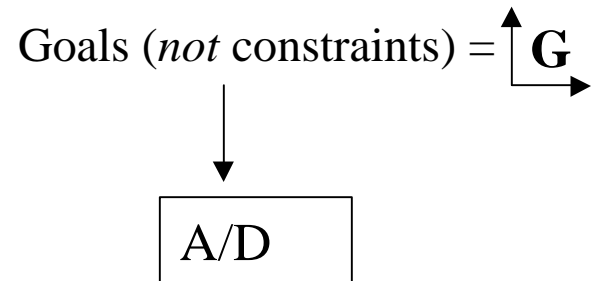# Bottom-up Design Methodology

- Advantages

  - Simple

- Disadvantages

  - A lot of wasted effort when "anticipated needs" of building blocks are wrong

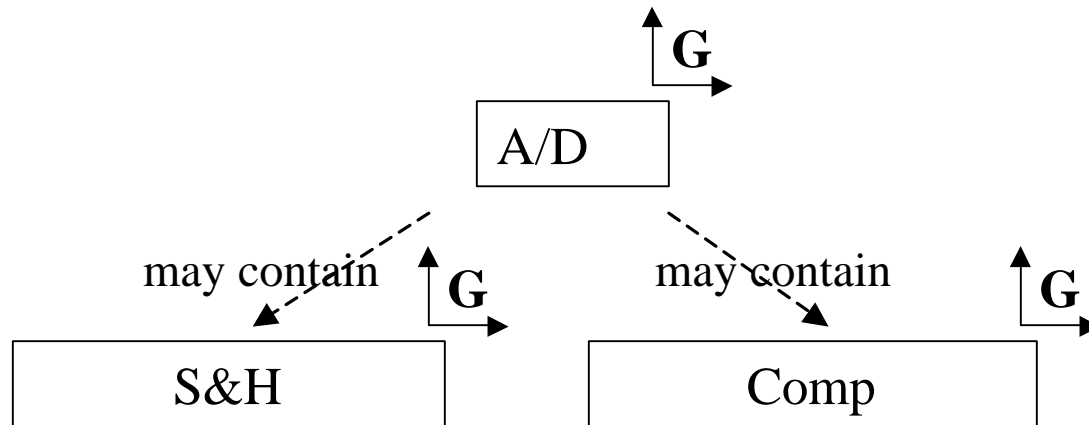 - Usually not rigorously structured, causing many iterations among levels in the hierarchy

Presentation Summary

- Introduction
- Review of Other Design Methodologies
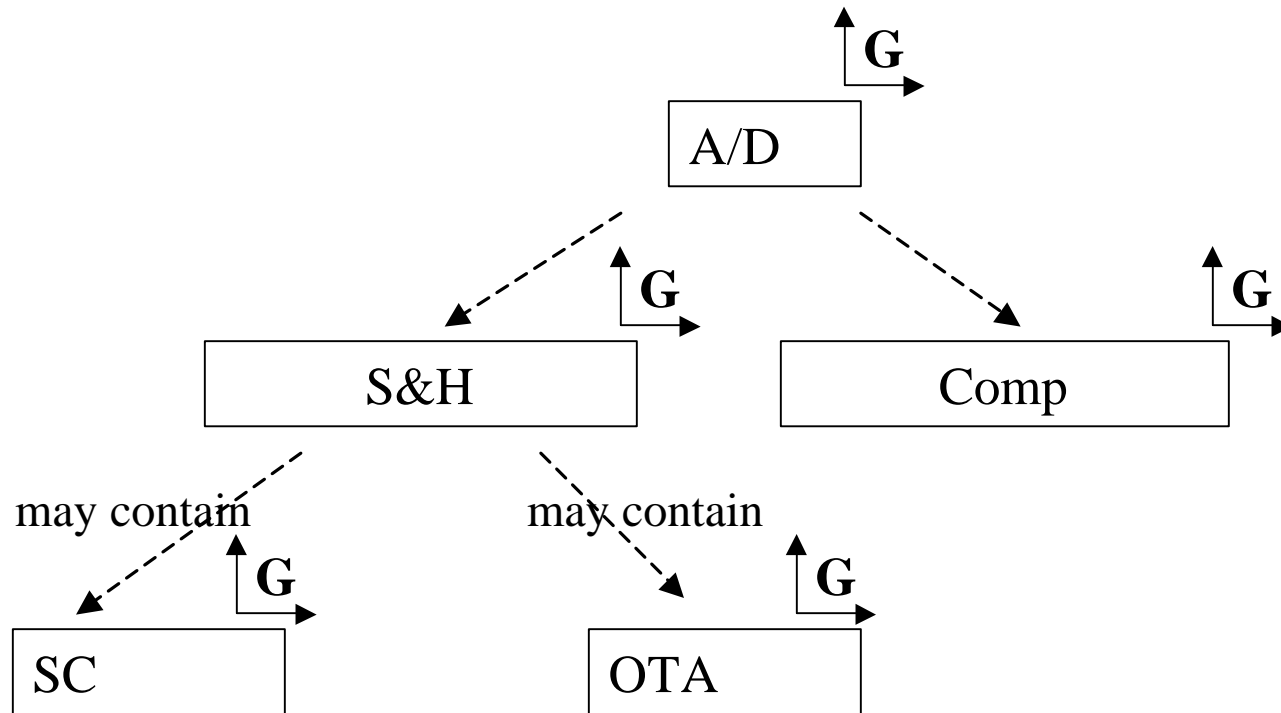- **Presentation of the TDBU**
- Ramifications of the TDBU
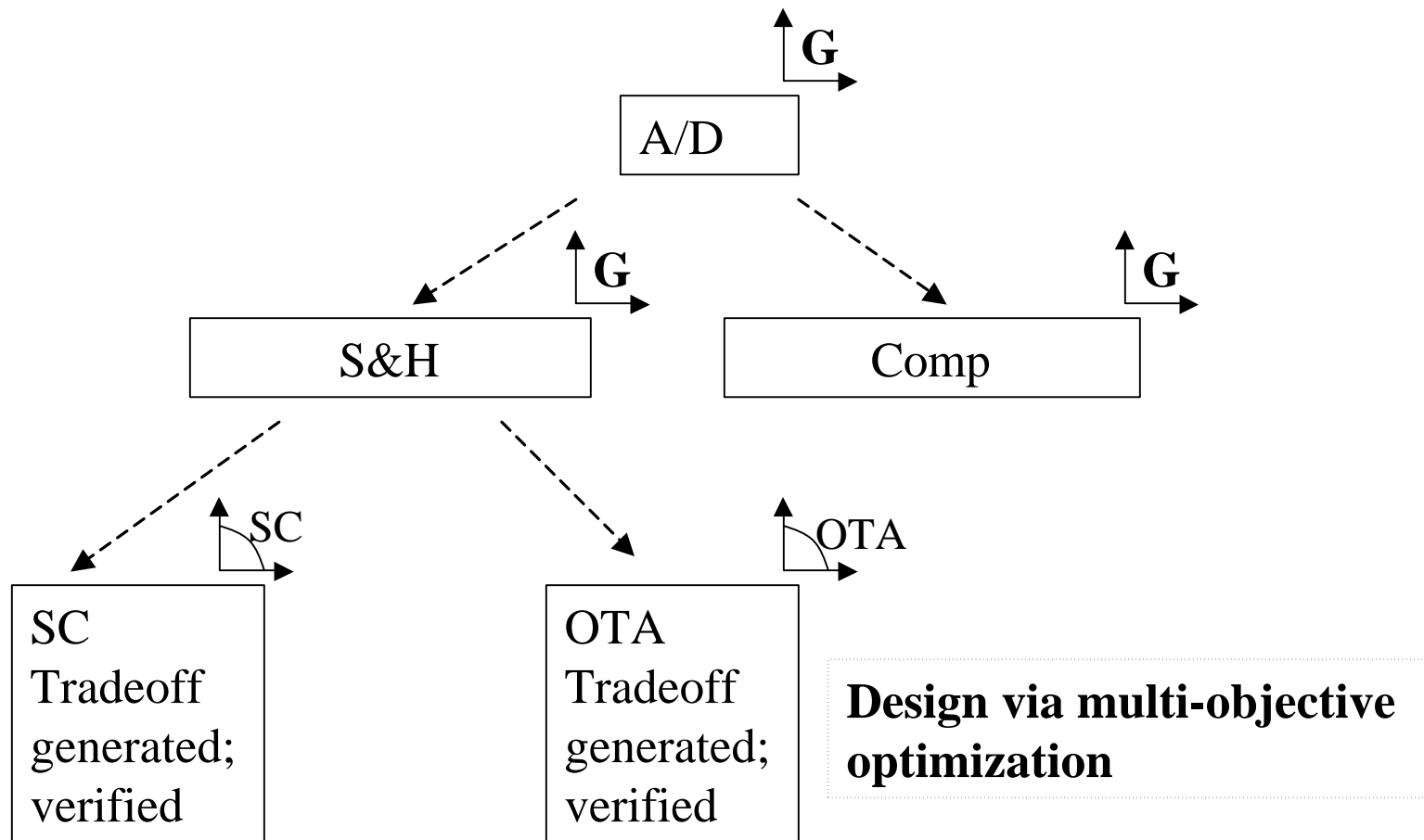- Conclusion

TDBU: Visualization

Goals (*not* constraints) = G
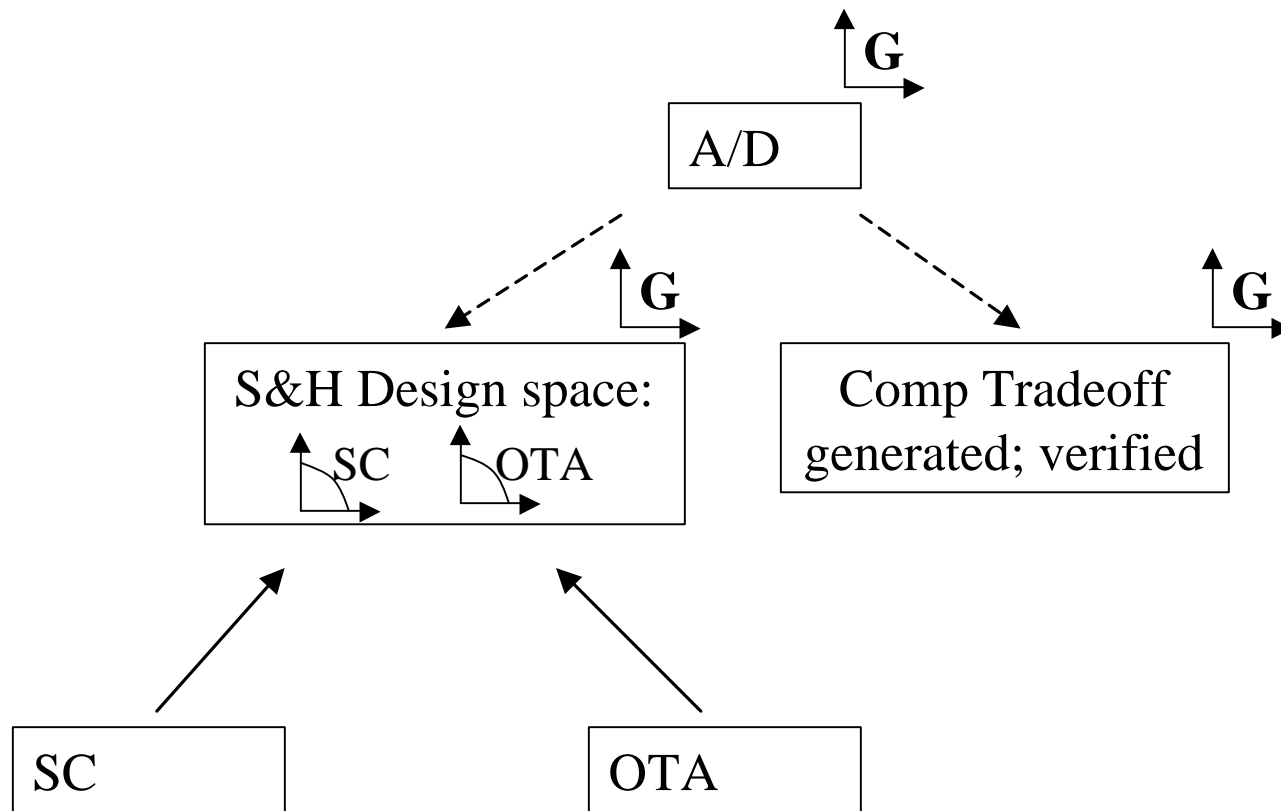
A/D

TDBU: Visualization
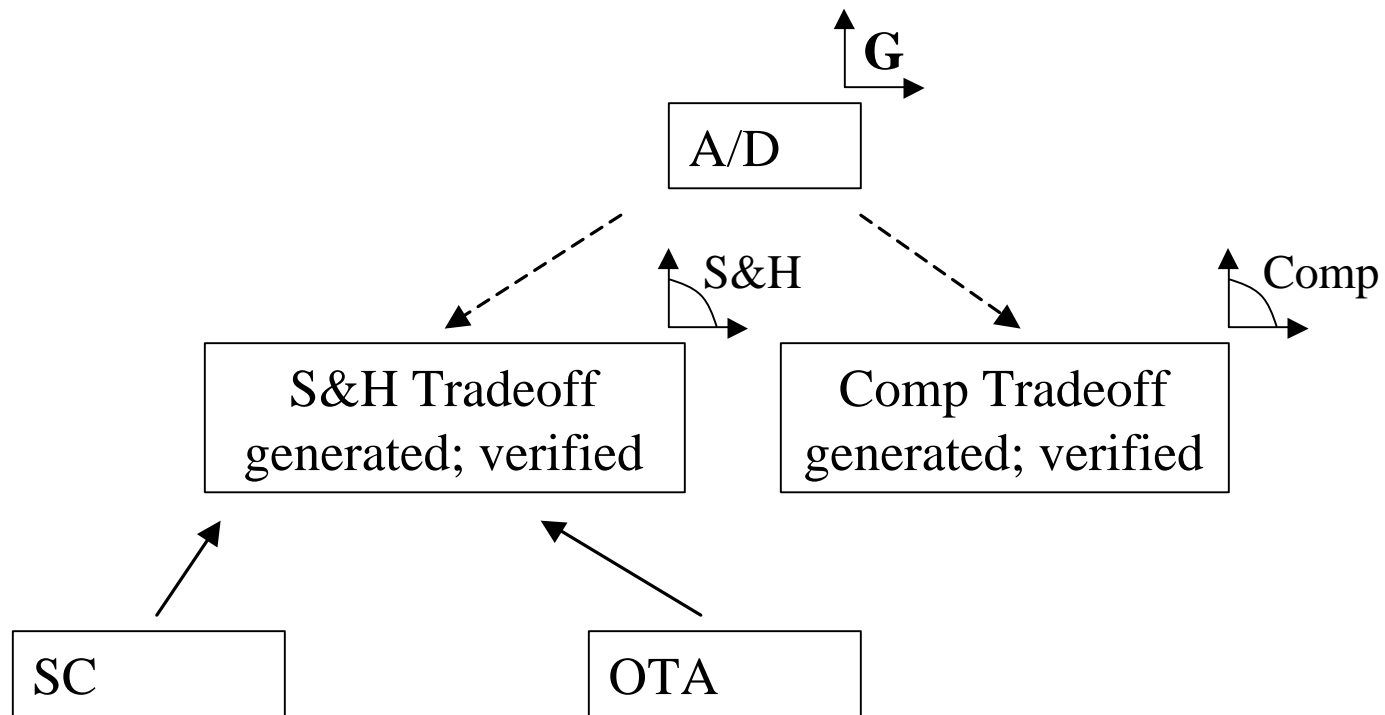
TDBU: Visualization
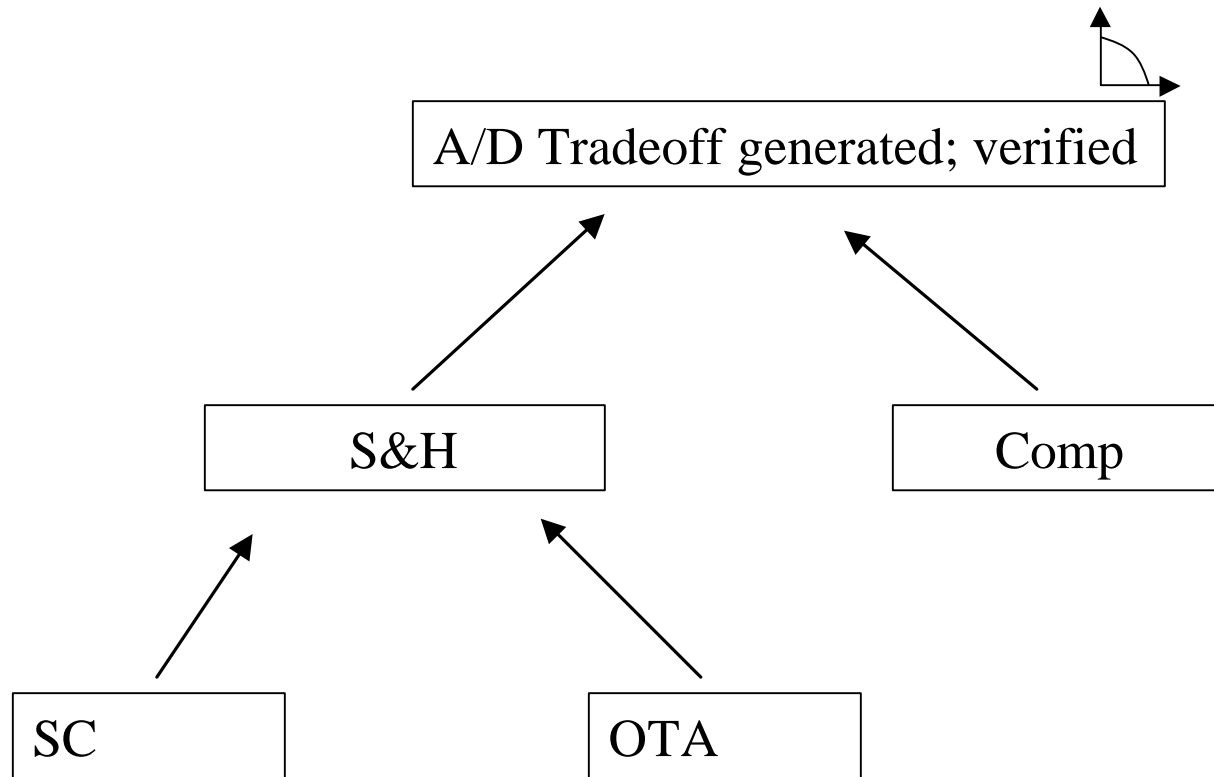
TDBU: Visualization

TDBU: Visualization

TDBU: Visualization

TDBU: Visualization

Presentation Summary

- Introduction
- Review of Other Design Methodologies
- What Makes a Good Methodology
- **Ramifications of the TDBU**
- Conclusion

Ramifications:
TDBU Has Useful Features

- Few iterations
- Provides optimal tradeoff curves for informed decision-making
- Hierarchical modeling of problem
- A tradeoff curve is an IP database encouraging reuse
- The bottom-up step can be parallelized
- General engineering methodology

Ramifications: Benefits of the TDBU

- Can handle massive complexity
- Minimizes design time
- Minimizes number of people needed
- Maximizes design quality – optimal results

Presentation Summary

- Introduction
- Review of Other Design Methodologies
- What Makes a Good Methodology
- Ramifications of the TDBU
- **Conclusion**

TDBU: Conclusion

- The goal is to design *complex* circuits *quickly*
- Conventional methodologies have major problems
- TDBU is a new methodology that overcomes these problems
- Can be applied to many problems in EH; EH solutions can automate parts of this methodology